第六章 網路管理

6-1 網路環境規劃

6-1-1 網路設定範例

圖 6-1 為一般學校網路架構，它透過中華電信數據專線(或 ADSL 網路)連結到 TANet 網路中心(譬如中山大學)。進入校園網路之前，經過外部路由器(含防火牆)連結道內部網路，TANet 是屬於 B Class 網路(255.255.0.0)，網路編號是由前兩個數字組構成，譬如範例校園網路編號是 120.118.0.0。也就是說，TANet 網路中心(如中山大學)會將網路號碼為 120.118.0.0 的封包轉送到該校園網路內，該校園內各個主機的 IP 位址必須設定在 120.118.0.1 ~ 120.118.255.255 之間才收得到網路訊息。理論上 B Class 網路的後兩碼是主機位址，但學校內有許多單位，期望每一單位有獨立的子網路，因此將第三碼再設定為子網路號碼，因此 IP 遮罩(IP Mask)就成為 255.255.255.0，子網路範圍由 120.118.0.* ~ 120.118.254.*。

接著，在校園網路內，利用某部路由器將 120.118.167.0 網路訊息轉送到該路由器的某一埠口(或稱網路介面)，該埠口所連接出去的網路範圍就介於 120.118.167.1 ~ 120.118.167.254，一般都會將該部口得 IP 位址設定在最後面，即是 120.118.167.254，又該部口是此網路進出外部網路的主要閘門，又稱為預定夾門(Default Gateway)。

依照此規劃，某一子網路可以有 1 ~ 253 部主機連結，一般都會利用網路交換器(Switch)或集線器(Hub) 分配連結。每一部 Switch/Hub 最多只有 24 個埠口，每一埠口連結一部主機設備，如此可能需要多埠 Switch/Hub 堆疊起來才夠。Switch/Hub 僅具有封包轉送或廣播的功能，並不具有路由選擇分配的功能，因此對網路 IP 位址的劃分並不影響。

本書就依照圖 6-1 的網路設定範例，介紹如何規劃與管理網路環境。

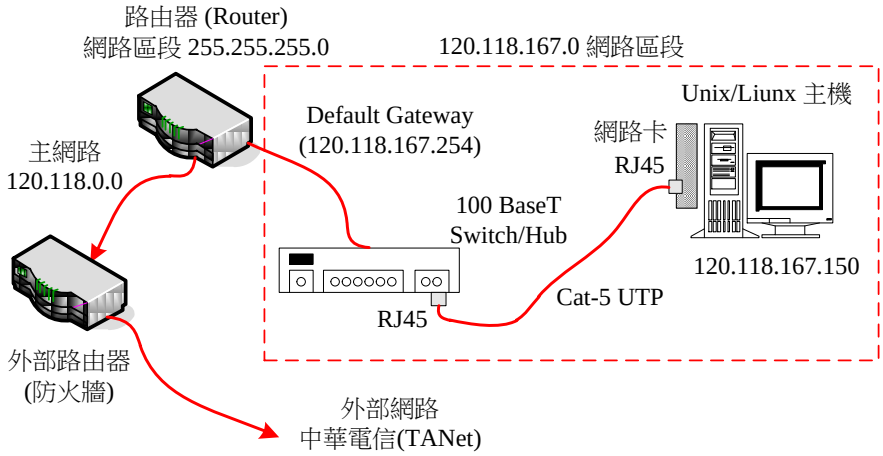


圖 6-1 網路設定範例

6-1-2 硬體裝置

下列是安裝網路基本的硬體裝置（如上圖 6-1 所示）：

- **網路卡：**目前以 100 Mbps、全雙工 Ethernet 網路卡最為普遍，有些可能已安裝於主機板內（如 acer 主機）。此外，也可安裝多片網路卡，系統會自動偵測網路卡，並針對每片網路卡給予獨立的識別名稱，如 eth0、eth1、等等；管理者則必須弄清楚每一片網路卡的識別名稱。

- **Cat-5 UTP 連線：**需要一條 Cat-5 UTP 的連線（100 Mbps 連線），其長度最長可達 100 米（現成的大多是 20 米），雙頭必須連接 RJ-45 接頭，一邊接網路卡，另一邊接 Switch/Hub。

- **Switch/Hub 集線器：**一般交換器（Switch）或集線器（Hub）皆有 8/16/24 個埠口，每一埠口可透過 Cat-5 UTP 連結一部主機或個人工作站。同時主機電腦也是透過 Switch/Hub 連結到其他網路設備（如 Router 或 Switch），再擴展到其他網路環境或外部網路。

6-1-3 網路環境選定

針對每一部主機電腦，必須設定下列網路環境：

- **IP 位址：**若是使用固定 IP，則必須選定其位址（需與原主機同一區段）；至於動態 IP(當工作站使用)，則不用選定，只需將組態設定成動態 IP 即可（由 DHCP 伺服器取得。）
- **IP 遮罩範圍：**設定 IP 位址中哪些是屬於網路號碼。一般 Class B 的 IP Mask 為 255.255.0.0，但如果經過 Subnet Mask 設定可就不一定了（本書範例為 255.255.255.0）。
- **DNS 伺服器位址：**設定服務該主機的 DNS 伺服器，一般組織單位（或自己網域內）都有專屬的 DNS 伺服器，如果沒有的話，可以設定 Hinet 的 DNS，其 IP 位址為 168.95.1.1。
- **主機名稱：**設定該主機名稱。

6-2 網路組態設定

一般 Unix/Linux 系統都有 ifconfig 供管理設定網路組態，另外許多系統也都有提供文字選單的操作工具 – setup，只要在一般文字終端機便可以操作。以下將利用這兩種工具分別介紹網路組態設定方法，同時介紹一相相關的設定檔。

6-2-1 介面命令設定 – ifconfig

利用 ifconfig 命令設定網路組態最為普遍，且無需任何輔助工具，步驟如下：

（A）設定 IP 位址：利用 ifconfig 設定 IP 位址格式如下：

```
# ifconfig 介面 IP-位址 [broadcast 廣播位址] [netmask 網路遮罩]
```

例如：（假設該網路卡的識別名稱為 eth0）

```
# ifconfig eth0 140.127.138.32 broadcast 140.127.138.255 netmask 255.255.255.0
```

其中介面和 IP 位址是必要的，廣播位址和網路遮罩部分，系統會依照網路等級（Class A ~ Class C）自動設定。但如有規劃次網路則必須指定廣播位址（如 140.127.138.255）和網路遮罩（255.255.255.0）。

（B）啟動網路介面：設定完成之後，接著必須啟動該網路卡（eth0）使其正常運作，操作如下：

```
# ifconfig  eth0  up
```

（D）顯示網路運作情形：還是利用 ifconfig 命令觀察網路運作情形，操作如下：

```
# ifconfig
eth0        Link encap:Ethernet  HWaddr 00:00:E2:5B:4A:65
            inet addr:140.127.138.32  Bcast:140.127.138.255  Mask:255.255.255.0
            inet6 addr: fe80::200:e2ff:fe5b:4a65/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:49850 errors:0 dropped:0 overruns:0 frame:0
            TX packets:4462 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:29052757 (27.7 MiB)  TX bytes:2957039 (2.8 MiB)
.......
```

上述範例若是設定成功，可觀察到相關參數（eth0 介面），否則表示網路組態沒有設定成功。管理者必須尋找出問題發生之所在，通常都是命令下錯，不然極有可能是網路卡已損壞或沒有插好。

6-2-2 組態工具設定 – nmuti

吾人可透過 nmuti 介面工具來設定網路環境，它是屬於 NetworkManager.service 服務套件內的管理工具，可利用 service 命令檢視該服務是否已安裝啟動，如下：（執行 # service NetworkManager.service status 命令）

```
# service NetworkManager.service status
Redirecting to /bin/systemctl status  NetworkManager.service.service
● NetworkManager.service - Network Manager
   Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled; vendor preset:
enabled)
   Active: active (running) since 六 2017-02-04 09:29:07 CST; 9min ago
     Docs: man:NetworkManager(8)
 Main PID: 650 (NetworkManager)
   CGroup: /system.slice/NetworkManager.service
           └─650 /usr/sbin/NetworkManager --no-daemon
…
```

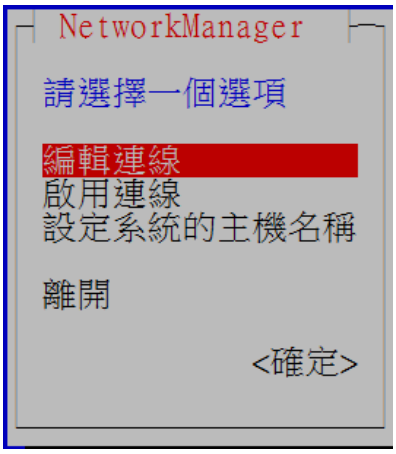
如果沒有請利用 yum 安裝，命令操作如下：(# yum –y install NetworkManager-tui)

```
# yum –y install NetworkManager-tui
Loaded plugins: fastestmirror, langpacks
base                                                    | 3.6 kB     00:00
extras                                                  | 3.4 kB     00:00
updates                                                 | 3.4 kB     00:00
updates/7/x86_64/primary_db                             | 2.2 MB     00:03
Loading mirror speeds from cached hostfile
 * base: mirrors.yun-idc.com
.....
```

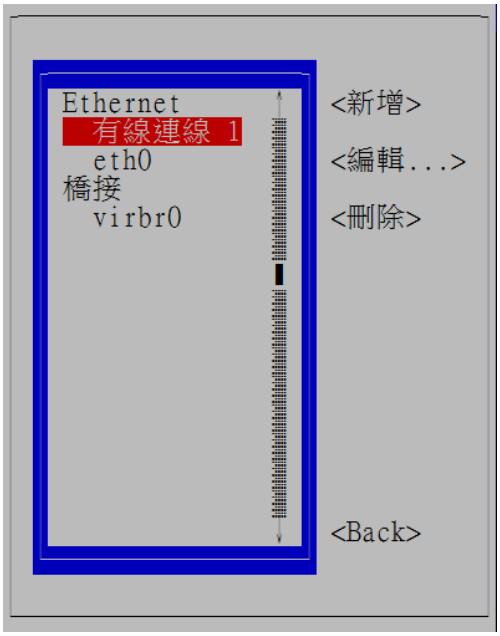
安裝後，終端機利用 nmtui 命令設定網路環，如下：(# nmtui 命令)

```
#nmtui
```

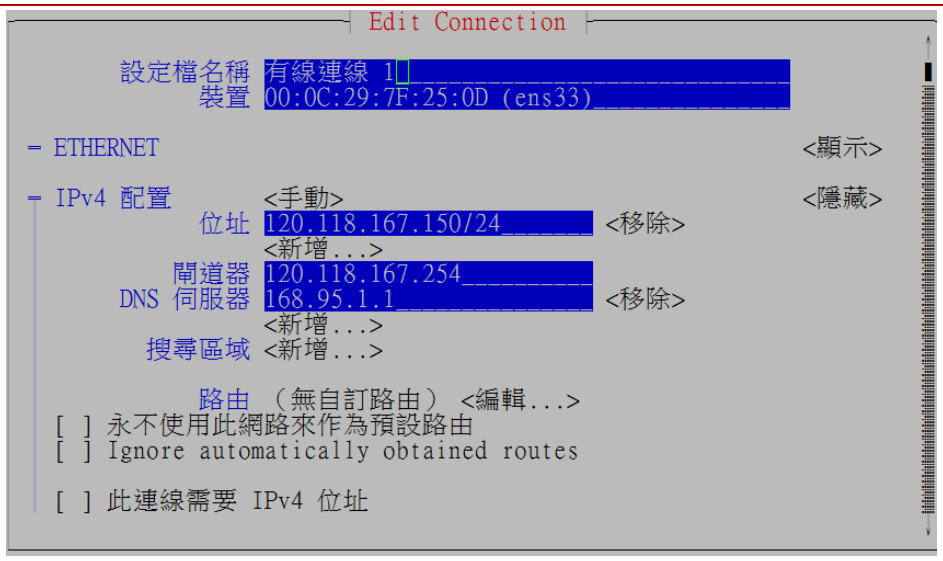
則會出現設定畫面，我們選擇『編輯連線』，再敲入 enter。



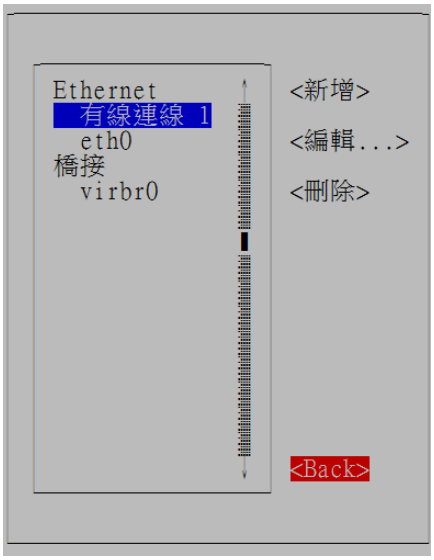
則出現下面視窗，它會出現兩片網路卡，其實都是同一只，選擇後按 Enter：



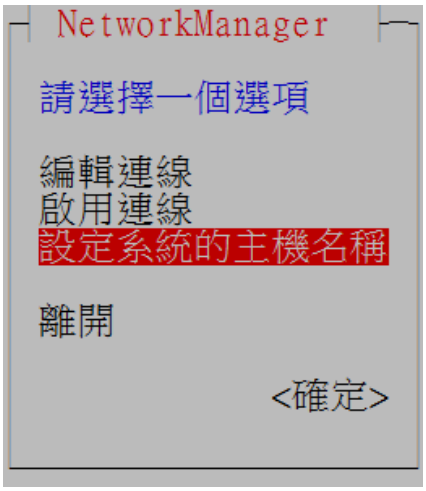
選擇介面卡之後，則依照您所規劃的網路環境輸入相關參數（但裝置名稱請不要修改），輸入完利用上下左右鍵，到最底下選確定。



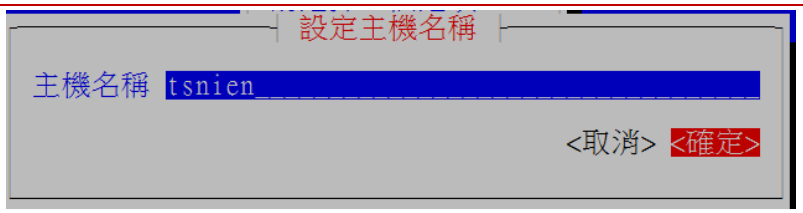
回到上一個視窗，選<Back>：



接著選『設定系統主機名稱』，如下：



輸入自己的主機名稱：



設定網路環境後，需重新啟動網路卡，如下：(# service NetworkManager restart)

```
# service NetworkManager restart
Redirecting to /bin/systemctl restart    NetworkManager.service
```

也可利用 ifconfig 觀察目前網路參數，如下：(#ifconfig 命令)

```
# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>    mtu 1500
            inet 120.118.167.150    netmask 255.255.255.0    broadcast 120.118.167.255
            inet6 fe80::1e18:1dd3:fce7:5e62    prefixlen 64    scopeid 0x20<link>
            ether 00:0c:29:7f:25:0d    txqueuelen 1000    (Ethernet)
            RX packets 33260    bytes 33196030 (31.6 MiB)
            RX errors 0    dropped 240    overruns 0    frame 0
            TX packets 8467    bytes 877036 (856.4 KiB)
            TX errors 0    dropped 0 overruns 0    carrier 0    collisions 0
....
```

6-3 網路組態檔案

早期 Unix/Linux 利用 /etc/sysconfig/network 設定網路組態，目前新的版本大多不理會此檔案，直接由 network-scripts 目錄下檔案與工具操作。

6-3-1 網路組態檔案 – /etc/sysconfig/network-scripts/

首先我們來觀察相關檔案(/etc/sysconfig/network-scripts/ 目錄下)，操作如下：

```
# cd /etc/sysconfig/network-scripts          (切換到網路設定檔目錄)
# ls                                         (查閱該目錄下檔案)
ifcfg-ens33
```

此檔案是針對網路介面組態的設定，一般經過網路介面與路由表設定後，就會將設定值寫入此檔案，無需再作修改，但也可以修改此檔案來取代設定。(查閱 ens33 網路介面為例)

```
# cat /etc/sysconfig/network-scripts/ifcfg-ens33      [執行 cat 命令]
```

```
TYPE=Ethernet
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
…
NAME=eth0
UUID=d5297909-ce0d-480f-b655-8626866f42e0
DEVICE=ens33
ONBOOT=yes
HWADDR=00:0C:29:65:71:B3
DNS1=168.95.1.1
IPADDR=120.118.167.150
PREFIX=24
GATEWAY=120.118.167.254
…
```

此檔案是針對網路介面組態的設定，一般經過網路介面與路由表設定後，就會將設定值寫入此檔案，無需再作修改，但也可以修改此檔案來取代設定。(查閱 ens33 網路介面為例)

```
# cat /etc/sysconfig/network-scripts/ifcfg-ens33      [執行 cat 命令]
TYPE=Ethernet
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
…
NAME=eth0
UUID=d5297909-ce0d-480f-b655-8626866f42e0
DEVICE=ens33
ONBOOT=yes
HWADDR=00:0C:29:65:71:B3
DNS1=168.95.1.1
IPADDR=120.118.167.150
PREFIX=24
GATEWAY=120.118.167.254
…
```

6-3-2 網路卡啟動/停止 – ifup/ifdown

吾人可利用 ifup 命令啟動如下：

```
# ifup    ens33        [執行啟動命令]
```

```
#
```

也可利用 ifdown 關閉網路卡：(終端機連線會中斷)

```
#ifdown    ens33        [執行關閉命令]

        [終端機連線中斷了，必須回主控台重新啟動]
```

6-4 網路相關檔案

6-4-1 主機 DNS 資料庫 - /etc/hosts

此檔案內存放較常使用的『IP 位址』和『主機名稱』對照表，當主機需要查詢主機的 IP 位址時，首先會到這個檔案搜尋，如果找不到再到網路上 DNS Server 上查詢 (目前幾乎沒有人在維護此檔案)。/etc/host 檔案範例如下：

```
$ cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       Linux-1.mis.csu.edu.tw Linux-1 localhost.localdomain localhost
```

每一行表示一筆主機名稱資料，其格式如下：

```
    IP 位址        主機 DNS 名稱        主機別名
```

6-4-2 主機服務埠口 - /etc/services

/etc/services 檔案記錄主機所提供的網路之『服務項目』、『埠口』(Port)、以及其所使用的『通訊協定』。基本上，1024 號以前的埠口都是固定給特定應用程式使用，因此，此檔案大多記載 1024 以前的埠口服務；但有些應用接在 1024 埠口以後，也會記錄在裡面，這也表示主機固定的使用埠口，都記錄在此檔案內，檔案範例如下 :(執行 # cat /etc/service 命令，節錄)

```
# /etc/services:
# Network services, Internet style
```

```
tcpmux          1/tcp                   # TCP port service multiplexer
tcpmux          1/udp                   # TCP port service multiplexer
echo            7/tcp
echo            7/udp
ftp            21/udp
ssh             22/tcp                  # SSH Remote Login
telnet         23/tcp
telnet         23/udp

domain          53/tcp          nameserver      # name-domain server
domain          53/udp          nameserver
```

6-4-3 TCP/IP 協定編號 - /etc/protocols

IP 封包內對於所承載的協定封包，都會給一個編號來識別所攜帶的通訊協定，這通訊協定號碼便登錄在 /etc/protocols 檔案內。該檔案範例如下：(執行 # cat /etc/protocols 命令，節錄)

```
ip          0           IP              # internet protocol, pseudo protocol number
#hopopt     0           HOPOPT          # hop-by-hop options for ipv6
icmp        1           ICMP            # internet control message protocol
igmp        2           IGMP            # internet group management protocol
ggp         3           GGP             # gateway-gateway protocol
tcp         6           TCP             # transmission control protocol
egp         8           EGP             # exterior gateway protocol
bbn-rcc 10          BBN-RCC-MON     # BBN RCC Monitoring
nvp         11          NVP-II          # Network Voice Protocol
pup         12          PUP             # PARC universal packet protocol
```

6-4-4 DNS 搜尋路徑 - /etc/host.conf

一般都是在本機 hosts (/etc/hosts) 上搜尋不到，再到 bind 主機 (DNS Server) 上搜尋。範例如下：(執行 # cat /etc/hosts.conf 命令)

```
multi on
```

6-4-5 DNS 搜尋順序 - /etc/resolv.conf

此檔案登錄主機尋找 DNS 伺服器的次序，以及 DNS 所在的 IP 位址。(執行 # cat

/etc/resolv.conf 命令)

```
# Generated by NetworkManager
nameserver 168.95.1.1
```

6-5 網路命令彙集

基本上，各種網路系統都有其專屬管理命令，我們在這裡僅介紹網路基本命令，分別說明如下。

6-5-1 測試網路介面 – ifconfig

ifconfig (Interface Configuration) 是設定網路組態最重要的命令，除了可以設定各類型的網路卡之外，也可顯示介面卡的訊息。首先，我們利用 ifconfig 來觀察網路卡的工作狀況，並瞭解網路卡的介面參數之後，再來利用 ifconfig 命令來規劃網路。ifconfig 命令格式如下：

```
ifconfig [interface]
ifconfig interface [aftype] options | address ...
```

利用 ifconfig 命令測試網路卡的工作狀況，如下： (執行 **#ifconfig** 命令)

```
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 120.118.165.120  netmask 255.255.255.0  broadcast 120.118.165.255
        inet6 fe80::1e18:1dd3:fce7:5e62  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:7f:25:0d  txqueuelen 1000  (Ethernet)
        RX packets 48623  bytes 34513511 (32.9 MiB)
        RX errors 0  dropped 612  overruns 0  frame 0
        TX packets 10017  bytes 2245849 (2.1 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 668  bytes 52084 (50.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 668  bytes 52084 (50.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

如果某個網路卡未被啟動，或安裝不成功(或未取得 DHCP 伺服器給予 IP 位址)工作，則該網路卡 (eth0、eth1) 的 IP 位址會看不到。由上述操作範例可以看出，eth0 與 lo 介面

已正常運作，也設有相關環境參數。各網路介面參數說明如下：

（A）網路介面型態

系統中常見的介面有下列幾種：

- 迴授（Loopback）介面 - lo

  此迴授介面為虛擬位址，即使沒有連結實際網路也有一個迴授介面。可當作一個網路介面來測試網路功能，資料由迴授介面送出，再送回主機。迴授介面名稱為 lo，而固定 IP 位址為 127.0.0.1。

- Ethernet 網路介面 – ens33

  如果主機上安裝有 Ethernet 網路卡，在開機時核心驅動程式就會自動尋找網路卡並建立介面。

- 點對點連線 – ppp0

  點對點連線是利用 PPP（Point-to-Point Protocol）通訊協定，以數據機連接電話撥接網路時使用。PPP 啟動時以 pppd 為 Daemon，第一個連接介面為 ppp0，依此類推。

（B）介面參數

我們以上述例子中顯示 eth0 和 lo 兩個介面的詳細資料，來說明有關網路介面的參數，如下：

- Hwaddr：Ethernet 網路位址。一般稱之為 MAC（Medial Access Control）位址，每一片網路卡都有其唯一固定的位址。

- inet addr：Internet 位址，亦是 IP 位址。如果是 PPP 介面，會由撥接的網路分配。

- Bcast：廣播位址。上例是 140.127.138.255。

- Mask：網路遮罩。亦是網路位址的範圍。
- MTU：最大傳輸單元（Maximum Transfer Unit）。表示這個介面不必分段，而能傳輸的最大封包，一般 Ethernet 介面為 1500 Bytes。
- Metric：向量值。RIP（Routing Information Protocol）用來計算路徑費用的向量值，該值愈大表示路徑成本愈高（例如，專線承租、或傳輸速率等等）。
- TX packets：傳送封包的總數、錯誤數量、遺失數量和溢流數量。
- RX packets：接收封包的總數、錯誤數量、遺失數量和溢流數量。

6-5-2 設定網路介面 - ifconfig

我們可利用 ifconfig 設定或更改介面卡參數，常用之設定格式如下：

```
# ifconfig 介面 IP-位址 [broadcast 廣播位址] [netmask 網路遮罩]
```

（A）設定 IP 位址

設定 IP 位址格式如下：

```
# ifconfig eth0 140.127.138.32 broadcast 140.127.138.255 netmask 255.255.255.0
```

其中介面和 IP 位址是必要的，廣播位址和網路遮罩部分，系統會依照網路等級（Class A ~ Class C）自動設定。但如果自行規劃次網路，則必須明確指定廣播位址（如 140.127.138.255）和網路遮罩（255.255.255.0）。

（B）停止或啟動網路介面

- 停止 eth0 網路介面運作

```
# ifconfig ens33 down
```

- 啟動 eth0 網路介面工作

```
# ifconfig ens33 up
```

6-5-3 靜態路由表設定 - route

對 Linux 核心而言，當要送出任一封包時，必須知道該封包應往哪一個路由傳送，因此，任一部主機都必須建構一個路由表（Routing Table），一般稱之為『靜態路由表』（Static Routing Table）。靜態路由的設定也是網路工作者最重要的工作，一般在 Linux 系統上是使用 route 命令。

（1）檢視靜態路由表

執行 /sbin/route 指令可以觀察系統核心內的路由表：(執行 # route 命令)

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         gateway         0.0.0.0         UG    100    0        0 ens33
120.118.167.0   0.0.0.0         255.255.255.0   U     100    0        0 ens33
192.168.122.0   0.0.0.0         255.255.255.0   U     0      0        0 virbr0
```

上述是由本書範例（圖 6-1）執行的結果，因只做主機系統（只有 ens33 網路卡）並沒有路由器功能，其路由表看起來會比較單調一點。每一行代表一個路徑選擇，各欄位功能說明如下：

- Destination：封包傳送的目的地之 IP 位址。
- Gateway：這條路徑所經過的網路閘門，其中 * 表示直接到達，而沒有經過網路閘門轉送。
- Gnemask：網路遮罩。
- Flags：表示這條路徑狀況之旗號，其意義為：
  - U 表示啟動（Up）。
  - H 表示這條路徑之目的地為主機（Host）。
  - G 表示這條路徑為網路閘門轉送（Gateway）。

- D 表示是經由 ICMP 重導路由設定（ICMP Redirect）。
- M 表示此路徑已經由修改（Modify）。

- Metric：此路徑之路由值。
- Refcnt：其他路徑經過的次數。
- Use：此路徑被使用的次數。
- Iface：這條路徑所經過的網路介面。

由上例中的最後一列 default，表示主機所欲傳送之目的位址不在路由表內，就傳送到 default 這條路徑上，通常 default 都是經由 Gateway 轉送。

（2）設定靜態路由表 - 主機為目的地

設定某一主機為路由的目的地之格式為：

```
# /sbin/route add –host [gw gateway] [metric cost] [netmask 遮罩] [dev 介面]
```

其中 metric、網路遮罩與介面通常都不用加入，系統會自行設定，例如：

```
# route add -host 140.127.138.33 eth0          [加入主機位址]
#   route                                      [顯示路由表內容]
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
140.127.138.33  *               255.255.255.255 UH    0      0        0 eth0
140.127.138.0   *               255.255.255.0   U     0      0        0 eth0
169.254.0.0     *               255.255.0.0     U     0      0        0 eth0
default         140.127.138.254 0.0.0.0         UG    0      0        0 eth0
```

（3）設定靜態路由表 – 以網路的路由

主機依照欲傳送之封包（或由其它介面進入之封包）的網路位址，來決定轉送到哪一個網路閘門，格式為：

```
# /sbin/route add –net 位址 [gw gateway] [metric cost] [netmask 遮罩] [dev 介面]
```

例如：

```
# route add -net 140.127.139.0 netmask 255.255.255.0 gw 140.127.138.254
                                    [增加一條路由表路徑]
# route                             [顯示路由表內容]
Kernel IP routing table
Destination     Gateway    Genmask          Flags Metric Ref    Use Iface
140.127.138.33  *          255.255.255.255 UH    0      0        0 eth0
140.127.139.0   140.127.138.254 255.255.255.0   UG    0      0        0 eth0
140.127.138.0   *          255.255.255.0   U     0      0        0 eth0
169.254.0.0     *          255.255.0.0     U     0      0        0 eth0
default         140.127.138.254 0.0.0.0     UG    0      0        0 eth0
```

（4）設定預設路由

預設路由（default）通常是通往網路外部的網路閘門。格式為：

```
# /sbin/route add default gw gateway_位址
```

例如：

```
# route add default gw 140.127.138.253          [加入預定匣門]
# route                                         [顯示路由表內容]
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
140.127.138.33  *               255.255.255.255 UH    0      0        0 eth0
140.127.139.0   140.127.138.254 255.255.255.0   UG    0      0 0 eth0
140.127.138.0   *               255.255.255.0   U     0      0        0 eth0
169.254.0.0     *               255.255.0.0     U     0      0        0 eth0
default         140.127.138.253 0.0.0.0         UG    0      0        0 eth0
default         140.127.138.254 0.0.0.0         UG    0      0        0 eth0
```

（5）刪除路由

刪除不必要的路由，例如：

```
# route del default gw 140.127.138.253
# route del 140.127.138.33
```

6-5-5 測試網路 – ping

ping 為最常用的網路測試命令，一般都是針對主機或網路閘門的 IP 位址進行測試，它

的方法是送出一個 ICMP Echo Request 封包到目的地，目的主機收到後會立即回覆一個 ICMP Echo Reply 給原發送端，由發送端計算來回的時間，並顯示出來。使用者利用此來回時間可以預估網路狀態，如果遺失封包過多或延遲時間過長表示網路狀態不穩定。但是當網路不正常，或主機不存在時，發送端在逾時（Time out）沒有收到 Echo Reply 時，也會顯示出來，並繼續發送 Echo Request 封包。ping 命令格式如下：

```
# ping  [選項]  主機位址
```

其實 ping 命令格式變化很多，我們僅列出較常用的選項：

- -c count：設定 ping 的傳送次數（count），預設值為 30。
- -d ：啟動 Socket 的 SO_DEBUG 功能。
- -f ：極速偵測（Flood ping），會快速的連續傳送 ping 封包來測試。
- -i wait：每筆偵測的相隔時間，預設值為 1 秒。
- -I Interface_address：發出封包的介面卡位址。
- -r ：忽略 Routing Table，而將封包直接送到遠端主機。。
- -R：記錄路由過程。
- -v ：顯示詳細的執行過程。

操作範例：(執行 # /bin/ping 168.95.1.1 命令)

```
PING 168.95.1.1 (168.95.1.1) from 192.168.0.50 : 56(84) bytes of data.
64 bytes from dns.hinet.net (168.95.1.1): icmp_seq=0 ttl=246 time=85.3 ms
--- 168.95.1.1 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 80.4/114.0/272.7 ms
```

一般主機發送 ICMP 封包時都將 TTL（Time-to-Live）設定為 255，因此由上例中，可以看出每一個封包回來時 TTL 皆為 246，表示到達目的主機（168.95.1.1）必須經過 9（255-246）個網路閘門，另外每個封包來回的時間也不一定相同。

6-5-6 路由追蹤 – traceroute

在網路上還有一個重要的管理命令就是 traceroute，當我們希望瞭解封包到達目的地的路徑狀況時，就必須利用 traceroute 來追蹤封包所經過的路徑。但在一般 Internet 網路上，每次封包所經過的路徑也許不同，因此，任何時間執行 traceroute 的結果當然就不會相同，首先我們來探討 traceroute 的運作原理。

最主要的是 traceroute 使用 ICMP 及 IP 標頭裡的 TTL（Time-to-Live）欄位。一般情況下，封包每經過一個網路閘門 TTL 值就被減 1，如果路由器收到一個 TTL 值減 1 以後為 0 時，便會回送一個 ICMP Time Exceeded（Type 11）(逾時）給原發送端，並將該封包丟棄，traceroute 就是利用這種特性來追蹤路徑。它的運作情況如下：首先 traceroute 送出一個 TTL 為 1 的 IP 封包到目的主機，第一個收到的路由器將 TTL 減 1，丟棄該封包，並回送 ICMP 給原發送主機，這個過程確認了這條路徑的第一個路由器；接下來，traceroute 再送 TTL 為 2 的 IP 封包，又可以得到第二個路由器位址，如此重覆一直到封包到達目的主機為止。但當封包到達目的位址時，它的 TTL 也被減成 0，同樣回應 ICMP Time Exceeded 封包，發送端如何來判斷封包已到達目的呢？

其實 traceroute 是以 UDP 封包格式發送，我們只要將 UDP 埠口設定在不可能使用的埠口上即可，一般都會將它設定較大的值（如 30000）。當目的主機收到後，判斷是自己的 IP 位址，但無此埠口服務，便會回應一個 ICMP Port Unreachable（埠口無法到達）給發送端，發送端只要利用『ICMP 埠口無法到達』和『ICMP 逾時』就可判斷是否到達目的主機。一般封包的 TTL 欄位預設值為 255（如 ping），這可能造成 traceroute 的封包在網路上無窮的回繞，因此，traceroute 的 TTL 預設值為 30，表示最高可以追蹤 30 個經過的網路閘門，但可以設定改變其大小。

traceroute 的命令格式如下：

```
# /usr/sbin/traceroute [選項] 主機位址
```

同樣的，traceroute 的選項也是很多，我們還是僅列出比較常用的選項：

- -d ：使用 Socket 層級的除錯功能。
- -F ：設定 Don't Fragment 位元。
- -g Gateway：設定路由器位址，最多 8 個。
- -m count：設定最大 TTL 值，預設值為 30。
- -n ：使用 IP 位址，而不用主機名稱。
- -r ：忽略 Routing Table，直接將封包傳送到目的位址。
- -v ：顯示詳細執行過程。

操作範例：(執行 **# traceroute 168.95.1.1**)

```
Tracing route to dns.hinet.net [168.95.1.1]
over a maximum of 30 hops:

  1     2 ms    <10 ms     1 ms   163.15.2.62
  2     *        *        *       Request timed out.
  3    72 ms    72 ms    70 ms   202.145.93.254
  4    75 ms    72 ms    76 ms   ks.ttn.net [202.145.84.254]
  5    81 ms    82 ms    78 ms   p25545.wan145.ficnet [202.145.255.45]
  6    79 ms    81 ms    79 ms   210.243.127.4
  7    85 ms    78 ms    81 ms   twix2.ttn.net [202.145.255.98]
  8    83 ms    80 ms    81 ms   211.22.41.186
  9    82 ms    83 ms    79 ms   211.22.35.98
 10    80 ms    79 ms    84 ms   211.22.35.1
 11    81 ms    81 ms    80 ms   dns.hinet.net [168.95.1.1]

Trace complete.
```

由上例中可以看出，我們追蹤到 168.95.1.1 經過了 9 個網路閘門(或路由器)，traceroute 針對每一個路徑送出 3 個 IP 封包，所回應的時間也不一定相同。追蹤路徑為：第一次到達 163.15.2.62 有回應，但經由 163.15.2.62 往下一個路徑 (第二次發送) 連續 3 個封包都沒有回應，因此退回來，又找出第一個路徑是 202.145.93.254，得到回應，再經由 202.145.93.254

往下一個路徑（202.145.84.254）；連續下去一直到達目的位址 168.95.1.1，其中共經過 9 個網路閘門。

6-5-7 顯示網路狀態 - netstat

顯示網路狀態命令 – netstat 是一個非常實用的工具，不但可以顯示網路運作情形，也可顯示路由表，以及其它重要的訊息，命令格式如下：

```
# /bin/netstat  [選項]
```

較常用的選項如下：

- -a ：顯示目前所有連接的 Socket。
- -c ：持續列出網路狀態。
- -C ：顯示路由快取資訊。
- -e ：顯示詳細的網路資訊。
- -g ：顯示多重廣播群組名單。
- -i ：顯示網路介面資訊。
- -I ：顯示傾聽中的 Socket 清單。
- -M ：顯示偽裝連線（NAT）。
- -n ：顯示 IP 位址而不是主機名稱。
- -r ：顯示路由表。
- -s ：顯示網路資訊統計清單。
- -t ：顯示 TCP 連線狀態。
- -u ：顯示 UDP 連線狀態。
- -v ：顯示詳細資訊。
- -w ：顯示 Raw Socket 連線。

操作範例：(執行 # netstat –rn 命令，查閱路由表)

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
140.127.139.0   140.127.138.254 255.255.255.0   UG        0 0          0 eth0
140.127.138.0   0.0.0.0         255.255.255.0   U         0 0          0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 eth0
0.0.0.0         140.127.138.254 0.0.0.0         UG        0 0          0 eth0
```

6-5-8 ARP 快取表命令 - arp

如果需要查詢或增減 ARP 快取表 (ARP Cache Table)，可以使用 arp 指令，格式為：

```
# /sbin/arp   [選項]
```

其中常用之選項為：

- -a ：顯示所有記錄。
- -d hostname：刪除 hostname 主機的記錄。
- -s hostname hw_address：增加 hostname 主機的硬體位址 (hw_address)

操作範例：(執行 # arp 命令)

```
Address                  HWtype  HWaddress           Flags Mask            Iface
gateway                  ether   00:25:46:86:88:4b   C                     ens33
120.118.165.107          ether   00:25:b3:0a:c1:17   C                     ens33
```

6-5-9 DNS 查詢 - nslookup

我們可以利用 nslookup 命令來查詢所指定 DNS 伺服器 (/etc/resolv.conf) 上的資料，也可以變換 DNS 伺服器查詢其他網域名稱的資料，如下：(執行 nslookup 命令)

```
> server                                    【目前指定的 DNS Server】
Default server: 140.127.1.1
Address: 140.127.1.1#53
> server 168.95.1.1                         【變換 DNS Server】
Default server: 168.95.1.1
Address: 168.95.1.1#53
```

```
> www.nsysu.edu.tw                    【查詢主機的 IP 位址】
Server:         168.95.1.1
Address:        168.95.1.1#53

Non-authoritative answer:
Name:    www.nsysu.edu.tw
Address: 140.117.11.112
> exit                                【離開 nslookup】
```

6-6 防火牆設定 – CentOS 7

6-6-1 防火牆運作程序

『防火牆』(Firewall) 是公眾網路與私有網路之間的通道隘口，功能是過濾封包進出。簡單的說，防火牆好比是城門的防護措施，如果防護太過嚴密 (甚至關閉城門)，便會失去建構網路的目的；但過於鬆散，易使內部資料暴露於外人之手，其間實難取捨。一般就安全措施的鬆緊度而言，主要依照私有網路的『安全政策』(Security Policy) 而定，並沒有一定的標準。

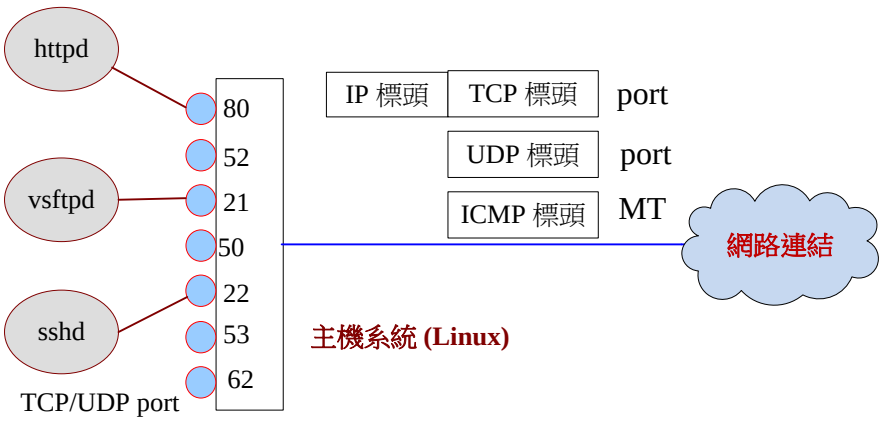


圖 6-2 主機防火牆的功能

我們可用圖 6-2 來說明防火牆的功能。防火牆是介於公眾網路和私有網路 (或稱內部網路、受保護的網路) 之間，是所有對內/對外通訊的『咽喉點』。當外部網路使用者欲傳送訊息進入內部網路，稱為『進入』(Inbound) 封包；而內部使用者送往外部網路的訊息，則

稱為『外出』(Outbound) 封包。防火牆功能就是管制『進入』與『外出』封包的進出，以達到安全防護的目的。

『封包過濾』(Packet Filtering) 是防火牆最基本的功能，它檢視進出封包是否符合安全規範，再決定是否給予放行。好像很困難其實很簡單，目前 Internet 網路大多採用 TCP/IP 網路協定，任何服務通道都以 TCP 或 UDP 埠口為管道，任何網路服務都掛在 TCP/UDP 埠口上，只要管理這些埠口就可以限制服務是否開放。另一方面，網路控制訊息大多採用 ICMP 封包傳送，如果管制某些 ICMP 進出，就可以限制外部探索私有網路。因此，我們歸納防火牆的封包過濾有限列途徑：

(1) IP/TCP 封包過濾：由封包的 IP 與 TCP 標頭某些欄位決定是否給予通行。

(2) IP/UDP 封包過濾：由封包上 IP 與 UDP 標頭某些欄位決定是否給予通過。

(3) IP/ICMP 封包過濾：由封包上 IP 與 ICMP 標頭某些欄位決定是否給予通過。

基本上，我們會將主機上所有 TCP 埠口、UDP 埠口與 ICMP 訊息**全部關閉**，當需要開啟某些服務(或伺服器，如 httpd、ftpd、sshd)時，再開啟相對應的埠口。

6-6-2 防火牆套件安裝與命令

CentOS 7 防火牆管理已不再採用 iptable 套件，而更新為 firewalld 套件，接下介紹 firewalld 的安裝與管理。

(A) 觀察 firewalld 是否安裝啟動

查詢 firewalld 是否安裝命令如下：

```
[root@secureLab ~]# rpm -qa | grep firewalld
firewalld-filesystem-0.8.2-2.el8.noarch
firewalld-0.8.2-2.el8.noarch
```

一般 Linux 系統安裝時，大多會自動安裝 firewalld 套件，如果沒有的話，則要自行安裝。

(B) 安裝 firewalld 套件

Firewalld 安裝與啟動命命操作如下：

```
# yum -y install  firewalld    【安裝 firewalld 套件】
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: ftp.isu.edu.tw
 * epel: mirror01.idc.hinet.net
 * extras: ftp.isu.edu.tw
 * updates: ftp.isu.edu.tw
…..
Complete!

# systemctl  start  firewalld          【啟動 firewalld】
# systemctl  enable firewalld          【設定開機時啟動 firewalld】
# systemctl  status  firewalld         【觀察 firewalld 執行狀況】
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset:
enabled)
   Active: active (running) since 四 2017-04-13 10:33:12 CST; 23s ago
     Docs: man:firewalld(1)
 …..
```

(C) Firewalld 相關命令

功能	命 令 格 式
安裝套件	# yum –y install firewalld firewalld-config
啟動	#systemctl start firewalld
停止	#systemctl stop firewalld
重新啟動	#systemctl restart firewalld
設定開機啟動	#syatemctl enable firewalld
查看狀態	#syatemctl status firewalld
更新防火牆規則	#firewall-cmd --reload
查看選定 zone	#firewall-cmd -- get-default-zone
查看 zone 開啟服務	#firewall-cmd --zone=public --list-all

查看永久開啟服務	#firewall-cmd --list-all --permanent
開啟 80 埠口固定	#firewall-cmd --add-port=80/tcp --permanent
關閉 80 埠口固定	#firewall-cmd --remove-port=80/tcp --permanent
開啟網路區段	#firewall-cmd --add-source=180.118.164.0/24
移除網路區段	#firewall-cmd --remove-source=180.118.164.0/24
開啟服務	#firewall-cmd --add-service=http --permanent
關閉服務	#firewall-cmd --remove-service=http --permanent
關閉 DHCP 服務	#firewall-cmd --remove-service dhcpv6-client

6-6-3 開啟相關伺服器埠口

吾人希望在主機上開啟 httpd(80/tcp)與 ftpd(20/tcp、21/tcp、22/tcp) 服務，操作如下：

```
[root@serCourse ~]# firewall-cmd --list-all --permanent 【查詢目前開啟，沒有】
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client ssh      [已開啟 ssh]
  ports:
  protocols:
  masquerade: no
  forward-ports:
  sourceports:
  icmp-blocks:
  rich rules:
[root@serCourse ~]# firewall-cmd --add-port=80/tcp --permanent 【開啟 httpd】
Success          【成功】
[root@serCourse ~]# firewall-cmd --add-port=20/tcp --permanent 【開啟 vsftpd】
Success      【成功】
[root@serCourse ~]# firewall-cmd --add-port=21/tcp --permanent 【開啟 vsftpd】
Success      【成功】
[root@serCourse ~]# firewall-cmd --add-port=22/tcp --permanent 【開啟 vsftpd】
Success      【成功】
#
```